



**Laboratoire de l'Informatique du Parallélisme**

École Normale Supérieure de Lyon  
Unité Mixte de Recherche CNRS-INRIA-ENS LYON-UCBL n° 5668

## *Up-to Techniques for Weak Bisimulation*

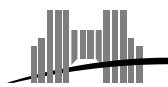
Damien Pous

Avril 2005

Research Report N° 2005-16

**École Normale Supérieure de Lyon**

46 Allée d'Italie, 69364 Lyon Cedex 07, France  
Téléphone : +33(0)4.72.72.80.37  
Télécopieur : +33(0)4.72.72.80.80  
Adresse électronique : [lip@ens-lyon.fr](mailto:lip@ens-lyon.fr)



# Up-to Techniques for Weak Bisimulation

Damien Pous

Avril 2005

## Abstract

Up-to techniques have been introduced to enhance the bisimulation proof method for establishing bisimilarity results. While up-to techniques for strong bisimilarity are well understood, in the weak case they come as a collection of unrelated results, and lack a unified presentation.

We propose a uniform and modular theory of up-to techniques for weak bisimulation that captures existing proof technology and introduces new techniques. Some proofs rely on non trivial – and new – commutation results based on termination guarantees.

The flexibility and usefulness of our framework is illustrated on two examples, one of them coming from a recent study of a distributed abstract machine.

**Keywords:** behavioural equivalence, bisimulation, termination, commutation diagrams

## Résumé

Les techniques *up-to* ont été introduites pour améliorer les méthodes coinductives (bisimulation) permettant de prouver des résultats de bisimilarité. Alors que les techniques up-to sont bien définies et comprises dans le cadre de la bisimilarité forte, elles se résument à un amoncellement de petits résultats, qui ne s'expriment pas dans un cadre uniforme dans le cas faible.

Nous proposons une théorie uniforme et modulaire des techniques up-to pour la bisimilarité faible qui capture les méthodes préexistantes, et en définit de nouvelles. Certaines de nos preuves exploitent des résultats non triviaux – dont un nouveau – de commutation, s'appuyant sur des arguments de terminaison. La flexibilité et l'utilité de notre théorie est illustrée avec deux exemples, dont l'un provient d'une étude récente d'une machine abstraite distribuée.

**Mots-clés:** équivalence comportementale, bisimulation, terminaison, commutation

## Introduction

*Bisimilarity* is a widely used behavioural relation in concurrency theory. It can be seen as the finest extensional equivalence that enjoys a natural formulation and nice mathematical properties. Bisimilarity can be defined as the greatest *bisimulation*. Given a *labelled transition system* (LTS), allowing one to write transitions between states of the form  $P \xrightarrow{\alpha} P'$  (meaning that a state  $P$  can perform an action  $\alpha$  and evolve to  $P'$ ), we say that a relation  $\mathcal{R}$  between states is a bisimulation whenever the leftmost diagram below holds: if  $P$  and  $Q$  are related by  $\mathcal{R}$  and  $P \xrightarrow{\alpha} P'$ , there is  $Q'$  such that  $Q \xrightarrow{\alpha} Q'$  and  $\mathcal{R}$  relates  $P'$  and  $Q'$ , and symmetrically for the transitions of  $Q$ .

$$\begin{array}{ccccc}
 P & \mathcal{R} & Q & & P & \mathcal{R} & Q & & P & \mathcal{R} & Q \\
 \alpha \downarrow & & \downarrow \alpha & & \alpha \downarrow & & \downarrow \alpha & & \alpha \downarrow & & \downarrow \alpha \\
 P' & \mathcal{R} & Q' & & P' & \mathcal{F}(\mathcal{R}) & Q' & & P' & \mathcal{S} & Q'
 \end{array}$$

Bisimulation is the most popular technique to establish bisimilarity between two processes: to prove that  $P$  and  $Q$  are bisimilar (written  $P \sim Q$ ), exhibit a bisimulation  $\mathcal{R}$  such that  $P \mathcal{R} Q$ . *Up-to techniques for bisimulation* have been introduced to alleviate the task of bisimulation proofs, by working with smaller relations. The proof scheme is shown on the second diagram above: a correct up-to technique is given by a function  $\mathcal{F}$  from relations to relations such that if we prove that  $\mathcal{R}$  ‘evolves to’  $\mathcal{F}(\mathcal{R})$ , then we know that  $\mathcal{R} \subseteq \sim$ . The advantage is that  $\mathcal{R}$  need not be a bisimulation (and can be ‘much smaller’ than a bisimulation). The notion of evolution of relations (depicted on the third diagram, where  $\mathcal{R}$  evolves to  $\mathcal{S}$  — its informal meaning is made precise below) serves as the basis of [7], where a general theory of up-to techniques for bisimulation is presented. The corresponding framework gives a unified and modular view of known up-to techniques, that can be combined together to yield powerful proof techniques for bisimilarity.

Up to now, we have implicitly been referring to the *strong* version of bisimulation. When analysing nontrivial systems, however, one is often interested in the *weak* version, where a special action, called  $\tau$ , is isolated, and the game of bisimulation is redefined by abstracting over  $\tau$  transitions ( $\tau$  is treated as a *silent* action, while other actions are *visible*). In the weak version of the bisimulation game, as shown on the rightmost diagram above,  $Q$  responds to  $P \xrightarrow{\alpha} P'$  by performing an  $\xrightarrow{\alpha}$  transition: this means that  $Q$  can do zero or several silent steps before and after the transition along  $\alpha$ , or even not move at all in the case where  $\alpha = \tau$  (and symmetrically when  $Q$  offers a challenge). One might then want to follow the same path as above: redefine the evolution of relations, and look for some functions  $\mathcal{F}$  that yield correct up-to proof techniques for weak bisimilarity (written  $\approx$ ). An important motivation for doing so is that in general, weak bisimulation proofs tend to be much larger than strong bisimulation proofs, so that having up-to techniques for the weak case is at least as important as in the strong case.

Unfortunately, in the weak case, irregularities appear, the paradigmatic example being given by the unsoundness of the ‘weak bisimulation up to weak bisimilarity’ proof technique. We recall the counterexample, from [8]. We suppose that the reader is familiar with CCS, and define  $\mathcal{R} \triangleq \{(\tau.a, 0)\}$ . Let us show that  $\mathcal{R}$  is a weak bisimulation up to  $\approx$ , i.e., that  $\mathcal{R}$  evolves to  $\approx \mathcal{R} \approx$  (we use juxtaposition to denote relation composition). The right process, 0, cannot

move. The only move the left process can do is a  $\tau$  transition to  $a$ , to which the right process answers by no move, and we get the pair  $(a, 0)$ . Now since we are reasoning up to  $\approx$ , and since  $a \approx \tau.a$ , we are allowed to replace this pair with  $(\tau.a, 0)$ , and we are back in  $\mathcal{R}$ . Nevertheless, we obviously cannot conclude that  $\tau.a$  and  $0$  are bisimilar processes.

Novel and useful proof techniques have been introduced to circumvent this difficulty [8, 3], notably based on the expansion preorder [1], that allows one to avoid situations where one can ‘undo a  $\tau$  transition’ as in the example above. However, as we have experienced in a recent study [4], in some cases reasoning up to expansion is not possible. We explain in Sect. 5 why expansion is not applicable in the setting of [4]. The intuitive reason can be formulated as follows: when a process  $P$  expands a process  $Q$ ,  $P$  has to be more efficient (in terms of internal computations, represented by silent transitions) than  $Q$  *at every step*. Typically, expansion is a well suited relation to get rid of intermediate computation steps that do not affect the behaviour of the system. However, it is common (in particular, it is the case in [4]) that along such transitions, an increased efficiency is achieved at the cost of some initial computation. Because of its ‘very controlled’ nature, expansion fails in handling this kind of pre-calculation techniques.

In the present work, we develop a theory of up-to techniques for weak bisimulation that enjoys nice properties in terms of generality and modularity, and we introduce new useful proof techniques for weak bisimilarity that can be used in that framework.

We start by adapting the work of [7] to the weak case, yielding the notion of *monotonic function* over relations. We explore the class of monotonic functions, and argue that it is too restrictive. We are thus led to relax the notion of monotonicity, and introduce *weakly monotonic* functions, for which up-to techniques can be applied only to reason about visible actions (those that cannot be undone by  $\approx$ ). We then show under which conditions monotonic and weakly monotonic functions can be combined together to obtain sound proof techniques. The resulting framework gives a unified and modular account of existing technology for weak bisimulation proofs. Beyond that, we validate some proof principles, such as ‘up to bisimilarity and transitivity on visible actions’, that to our knowledge had not been proposed before.

We then attack the question of finding alternatives to the expansion relation to handle  $\tau$  transitions in weak bisimulation proofs. We propose an *up to controlled simulation* technique. The notion of controlled simulation intuitively captures the idea of avoiding ‘going back in time’ in bisimulation proofs. We introduce *relaxed expansion*, a co-inductively defined relation that is a controlled simulation and is coarser than expansion. We also propose two new proof principles for which the control on  $\tau$  steps exploits a different kind of argument, based on termination guarantees. The corresponding correctness proofs are best formulated as rewriting results, that are technically difficult and may be of interest *per se*; we therefore describe them in that setting in a dedicated section. Like all other results in the paper, except those of Subsection 2.3 (about faithful contexts), they have been formally checked in the Coq proof assistant [6].

To illustrate the power and the versatility of our techniques, we develop two examples where the benefits of our framework can be shown in terms of size and modularity of the proofs. For the sake of conciseness, these examples are

somewhat simplified (one of these comes from the aforementioned study in [4]), but we hope that they demonstrate that our approach can be used in large bisimulation proofs about complex systems.

**Outline of the paper.** In Sect. 1, we introduce some necessary background and show where the approach of [7] breaks when adapted to the weak case. We develop our theory of up-to techniques for weak bisimulation in Sect. 2, introducing monotonic and weakly monotonic functions. In Sect. 3 we introduce controlled simulations and present new up-to techniques based on this notion. The correctness of some of these techniques is supported by the proofs given in Sect. 4, which are formulated in the setting of commutation results. Sect. 5 illustrates the use of our framework on two nontrivial examples. We give final remarks in Sect. 6.

## 1 The Problem of “Weak Bisimulation Up to”

### 1.1 Labelled Transition Systems, Relations, Evolution

We consider a labelled transition system (LTS)  $(\mathcal{P}, \mathcal{L}, \rightarrow)$ , with domain  $\mathcal{P}$ , labels or actions in  $\mathcal{L}$  and transition relation  $\rightarrow \subseteq \mathcal{P} \times \mathcal{L} \times \mathcal{P}$ . The elements of  $\mathcal{P}$  are called *processes* and are denoted by  $P, Q$ . We distinguish a *silent action*,  $\tau \in \mathcal{L}$ . We let  $\alpha, \beta$  (resp.  $a, b$ ) range over actions, in  $\mathcal{L}$  (resp. *visible actions*, in  $\mathcal{L} \setminus \{\tau\}$ ). We write  $P \xrightarrow{a} Q$  when  $(P, a, Q) \in \rightarrow$  (so that  $P \xrightarrow{a} Q$  stands for a transition of  $P$  along a visible action  $a$ ).

We let  $\mathcal{R}, \mathcal{S}, \mathcal{B}, \mathcal{E}$  range over binary relations (simply called *relations* in the sequel) on processes, and denote respectively by  $\mathcal{R}^+, \mathcal{R}^=, \mathcal{R}^*$  the transitive, reflexive, transitive and reflexive closure of the relation  $\mathcal{R}$ .  $P \mathcal{R} Q$  stands for  $(P, Q) \in \mathcal{R}$ . The composition of two relations  $\mathcal{R}$  and  $\mathcal{S}$ , written  $\mathcal{R}\mathcal{S}$ , is defined by  $\mathcal{R}\mathcal{S} \triangleq \{(P, Q) \text{ s.t. } P \mathcal{R} T \text{ and } T \mathcal{S} Q \text{ for some process } T\}$ . We will also need the inverse of a relation:  $\mathcal{R}^{-1} \triangleq \{(P, Q) \text{ s.t. } Q \mathcal{R} P\}$ .  $\mathcal{I}$  will denote the identity relation. We say that  $\mathcal{R}$  *contains*  $\mathcal{S}$  (alternatively, that  $\mathcal{S}$  is contained in  $\mathcal{R}$ ), written  $\mathcal{S} \subseteq \mathcal{R}$ , if  $P \mathcal{S} Q$  implies  $P \mathcal{R} Q$ . A relation  $\mathcal{R}$  *terminates* if there is no infinite sequence  $P_1, P_2 \dots$  such that  $\forall i, P_i \mathcal{R} P_{i+1}$ .

**Definition 1.1 (weak transitions)** *The weak transition relation, written  $\xrightarrow{*}$ , is defined by the reflexive transitive closure of  $\xrightarrow{\alpha}$  when  $\alpha = \tau$ , and the composition  $\xrightarrow{\alpha} \xrightarrow{\beta} \xrightarrow{\gamma}$  otherwise.*

**Definition 1.2 (evolution)** *Let  $\alpha$  be an action and  $\mathcal{R}, \mathcal{S}$  two relations.  $\mathcal{R}$   $\alpha$ -evolves to  $\mathcal{S}$ , if whenever  $P \mathcal{R} Q$ ,  $P \xrightarrow{\alpha} P'$  implies  $Q \xrightarrow{\alpha} Q'$  and  $P' \mathcal{S} Q'$  for some  $Q'$ . Given two relations  $\mathcal{R}$  and  $\mathcal{S}$ , we say that:*

- $\mathcal{R}$  evolves to  $\mathcal{S}$ , denoted by  $\mathcal{R} \rightarrow \mathcal{S}$ , if  $\mathcal{R}$   $\alpha$ -evolves to  $\mathcal{S}$  for all  $\alpha \in \mathcal{L}$ ,
- $\mathcal{R}$  evolves silently to  $\mathcal{S}$ , denoted by  $\mathcal{R} \rightarrow_{\tau} \mathcal{S}$ , if  $\mathcal{R}$   $\tau$ -evolves to  $\mathcal{S}$ ,
- $\mathcal{R}$  evolves visibly to  $\mathcal{S}$ , denoted by  $\mathcal{R} \rightarrow_{\neq \tau} \mathcal{S}$ , if  $\mathcal{R}$   $a$ -evolves to  $\mathcal{S}$  for all  $a \in \mathcal{L} \setminus \{\tau\}$ .

Our notion of evolution is the ‘asymmetric’ version of *progression* in [7]:  $\mathcal{R}$  progresses to  $\mathcal{S}$  in the sense of [7] iff  $\mathcal{R}$  evolves to  $\mathcal{S}$  and  $\mathcal{R}^{-1}$  evolves to  $\mathcal{S}^{-1}$ .

The following lemma will be useful in the proofs below.

**Lemma 1.3** 1. Let  $\mathcal{R}$  be a relation. If  $\mathcal{R} \succ \mathcal{R}$ ,  $P \mathcal{R} Q$  and  $P \xrightarrow{\tau} P'$ , then there is  $Q'$  such that  $Q \xrightarrow{\tau} Q'$  and  $P' \mathcal{R} Q'$ .

2. Let  $\mathcal{R}$  be a relation. If  $\mathcal{R} \succ \mathcal{R}$ ,  $P \mathcal{R} Q$  and  $P \xrightarrow{a} P'$ , then there is  $Q'$  such that  $Q \xrightarrow{a} Q'$  and  $P' \mathcal{R} Q'$ .

3. Let  $\alpha$  be an action, and  $(\mathcal{R}_i)_{i \in I}$ ,  $(\mathcal{S}_j)_{j \in J}$  two families of relations. If for all  $i$  there is  $j$  such that  $\mathcal{R}_i \succ \alpha \mathcal{S}_j$ , then  $\bigcup_{i \in I} \mathcal{R}_i \succ \alpha \bigcup_{j \in J} \mathcal{S}_j$ .

*Proof.* (1) is proved by induction on the derivation  $P \xrightarrow{\tau} P'$ . (2) and (3) are straightforward.  $\square$

In the following, we build a theory of up-to techniques to reason about simulations. This leads to simpler developments, and we show at the end of each section how to use the results to obtain proof techniques for bisimulation.

In the definition below, and in the remainder of the paper, we implicitly refer to *weak* relations. There are several equivalent definitions of bisimilarity. The following directly gives the standard way to prove a bisimilarity result between two processes  $P$  and  $Q$ : exhibit a bisimulation relation  $\mathcal{R}$  containing the pair  $(P, Q)$ .

**Definition 1.4 (simulation, bisimulation, expansion)** Let  $\mathcal{R}$  be a relation,  $\mathcal{R}$  is a simulation (resp. silent simulation) if  $\mathcal{R} \succ \mathcal{R}$  (resp.  $\mathcal{R} \succ \mathcal{R}$ ).  $\mathcal{R}$  is a bisimulation if  $\mathcal{R}$  and  $\mathcal{R}^{-1}$  are simulations. Two processes  $P$  and  $Q$  are bisimilar, written  $P \approx Q$ , if  $P \mathcal{R} Q$  for some bisimulation  $\mathcal{R}$ .

Expansion, denoted by  $\succsim$ , is the largest relation such that  $\succsim^{-1}$  is a simulation, and, whenever  $P \succsim Q$ ,

1.  $P \xrightarrow{\tau} P'$  implies  $Q \xrightarrow{\tau} Q'$  and  $P' \succsim Q'$  for some  $Q'$ , or  $P' \succsim Q$ ;
2.  $P \xrightarrow{a} P'$  implies  $Q \xrightarrow{a} Q'$  and  $P' \succsim Q'$  for some  $Q'$ .

## 1.2 The Difficulty in the Weak Case

We now adapt the theory of up-to techniques of [7] to the weak case, and show where the difficulties arise.

We let  $\mathcal{F}, \mathcal{G}$  range over *functions* from relations to relations. We say that  $\mathcal{F}$  contains  $\mathcal{G}$ , written  $\mathcal{G} \subseteq \mathcal{F}$ , if  $\mathcal{G}(\mathcal{R}) \subseteq \mathcal{F}(\mathcal{R})$  for any relation  $\mathcal{R}$ . Given a relation  $\mathcal{S}$ , we define *identity* ( $\mathcal{U}$ ), *constant-to- $\mathcal{S}$*  ( $\tilde{\mathcal{S}}$ ),  *$\mathcal{S}$ -left-chaining* ( $\mathcal{S}\bullet$ ) and  *$\mathcal{S}$ -right-chaining* ( $\bullet\mathcal{S}$ ) as follows:

$$\mathcal{U}(\mathcal{R}) \triangleq \mathcal{R} \quad \tilde{\mathcal{S}}(\mathcal{R}) \triangleq \mathcal{S} \quad \mathcal{S}\bullet(\mathcal{R}) \triangleq \mathcal{S}\mathcal{R} \quad \bullet\mathcal{S}(\mathcal{R}) \triangleq \mathcal{R}\mathcal{S}$$

We define four *constructors*, i.e. functions from functions to functions: *composition* ( $\circ$ ), *union* ( $\cup$ ), *iteration* ( $*$ ) and *chaining* ( $\frown$ ), as follows:

$$\begin{aligned} (\mathcal{F} \circ \mathcal{G})(\mathcal{R}) &\triangleq \mathcal{F}(\mathcal{G}(\mathcal{R})) & (\mathcal{F}^0)(\mathcal{R}) &\triangleq \mathcal{R} \\ (\mathcal{F} \cup \mathcal{G})(\mathcal{R}) &\triangleq \mathcal{F}(\mathcal{R}) \cup \mathcal{G}(\mathcal{R}) & (\mathcal{F}^{n+1})(\mathcal{R}) &\triangleq \mathcal{F}^n(\mathcal{R}) \cup \mathcal{F}(\mathcal{F}^n(\mathcal{R})) \\ (\mathcal{F} \frown \mathcal{G})(\mathcal{R}) &\triangleq \mathcal{F}(\mathcal{R})\mathcal{G}(\mathcal{R}) & (\mathcal{F}^*)(\mathcal{R}) &\triangleq \bigcup_{n \geq 0} \mathcal{F}^n(\mathcal{R}) \end{aligned}$$

**Definition 1.5 (monotonicity)** A function  $\mathcal{F}$  is monotonic if  $\mathcal{R} \subseteq \mathcal{S}$  entails  $\mathcal{F}(\mathcal{R}) \subseteq \mathcal{F}(\mathcal{S})$  and the following conditions hold:

$$(1) \begin{cases} \mathcal{R} \succ \mathcal{S} \\ \mathcal{R} \subseteq \mathcal{S} \end{cases} \Rightarrow \mathcal{F}(\mathcal{R}) \succ \mathcal{F}(\mathcal{S}) \quad (2) \begin{cases} \mathcal{R} \succ \mathcal{S} \\ \mathcal{R} \subseteq \mathcal{S} \end{cases} \Rightarrow \mathcal{F}(\mathcal{R}) \xrightarrow{v} \mathcal{F}(\mathcal{S})$$

This slightly strengthens the notion of *respectfulness* found in [7], in which the two kinds of transitions are treated uniformly. While the results of this section would hold using respectful functions, we will need this separation between silent and visible actions in Sect. 2.2.

**Proposition 1.6 (correctness of monotonic functions)** *Let  $\mathcal{F}$  be a monotonic function. If  $\mathcal{R} \mapsto \mathcal{F}(\mathcal{R})$ , then  $\mathcal{F}^*(\mathcal{R})$  is a simulation.*

*Proof.* We show by induction that  $\mathcal{F}^n(\mathcal{R}) \mapsto \mathcal{F}^{n+1}(\mathcal{R})$  and conclude using Lemma 1.3(3).  $\square$

This proposition ensures that a monotonic function provides a sound up-to technique: whenever we can prove that  $\mathcal{R}$  evolves to  $\mathcal{F}(\mathcal{R})$ , then  $\mathcal{R}$  is contained in  $\mathcal{F}^*(\mathcal{R})$ , which is a simulation. We now exhibit some monotonic functions, and show how to combine them to obtain more powerful techniques.

**Lemma 1.7** *Let  $\mathcal{S}$  be a simulation,  $\mathcal{U}$ ,  $\tilde{\mathcal{S}}$ ,  $\bullet\mathcal{S}$  and  $\succsim\bullet$  are monotonic functions.*

In the sequel, we will say that a constructor *respects* a predicate  $P$  over functions, if, given arguments that satisfy  $P$ , it returns a function satisfying  $P$ .

**Lemma 1.8** *Composition ( $\circ$ ), union ( $\cup$ ) and iteration ( $*$ ) are constructors that respect monotonicity.*

We can now apply our framework to reason about bisimulation relations, and revisit a result from [8]. We show that the proof becomes elementary.

**Theorem 1.9** *If  $\mathcal{R} \mapsto \succsim \mathcal{R}^{\approx} \approx$  and  $\mathcal{R}^{-1} \mapsto \succsim (\mathcal{R}^{-1})^{\approx} \approx$ , then  $\mathcal{R} \subseteq \approx$ .*

*Proof.* Using the previous results,  $\mathcal{F}(\mathcal{R}) \triangleq \succsim \mathcal{R}^{\approx}$  is monotonic, and  $\mathcal{F}^*(\mathcal{R})$  and  $\mathcal{F}^*(\mathcal{R}^{-1})$  are simulations. Then  $\approx \mathcal{F}^*(\mathcal{R})$  and  $\mathcal{F}^*(\mathcal{R}^{-1}) \approx$  are simulations. We check that  $(\approx \mathcal{F}^*(\mathcal{R}))^{-1} = \mathcal{F}^*(\mathcal{R}^{-1}) \approx$ , so that  $\mathcal{R} \subseteq \approx \mathcal{F}^*(\mathcal{R}) \subseteq \approx$ .  $\square$

**The transitivity problem.** The  $\approx$ -left-chaining function is not monotonic. As a consequence, the chaining constructor does not respect monotonicity in general. To see why, let us try to prove the monotonicity of  $\approx\bullet$ . Given  $\mathcal{R}$  and  $\mathcal{S}$  such that  $\mathcal{R} \mapsto \mathcal{S}$  and  $\mathcal{R} \subseteq \mathcal{S}$ , we have to show  $\approx \mathcal{R} \mapsto \approx \mathcal{S}$ . This can lead to closing the leftmost diagram below. Our hypothesis allows us to close the first step of the transition  $P_1 \mapsto P'_1$ , and obtain the second diagram. However, from this point, we cannot do anything, since we have no hypothesis on the silent evolution of  $\mathcal{S}$ .

$$\begin{array}{ccccccc}
 P & \approx & P_1 & \mathcal{R} & Q & P & \approx & P_1 & \mathcal{R} & Q \\
 \tau \downarrow & & \tau \downarrow & & & \tau \downarrow & & \tau \downarrow & \mathcal{S} & \Downarrow \tau \\
 P' & \approx & P'_1 & & & P' & \approx & P'_1 & ? & Q_1
 \end{array}$$

## 2 A Smooth Theory for the Weak Case

### 2.1 A Weaker Notion of Monotonicity

When looking at the counterexample given in the Introduction, we can observe that the problem is related to silent transitions: unlike visible transitions, they can be cancelled by  $\approx$ . We now exploit this observation to relax the definition of monotonicity, which leads to a smoother theory, where reasoning *up to weak bisimilarity* is allowed, but on visible actions only.

**Definition 2.1 (weak monotonicity)** *A function  $\mathcal{F}$  is weakly monotonic if  $\mathcal{R} \subseteq \mathcal{S}$  entails  $\mathcal{F}(\mathcal{R}) \subseteq \mathcal{F}(\mathcal{S})$  and the following conditions hold:*

$$(1) \mathcal{R} \rightsquigarrow \mathcal{R} \Rightarrow \mathcal{F}(\mathcal{R}) \rightsquigarrow \mathcal{F}(\mathcal{R}) \quad (2) \left\{ \begin{array}{l} \mathcal{R} \rightsquigarrow \mathcal{R}, \quad \mathcal{R} \rightsquigarrow \mathcal{S} \\ \mathcal{S} \rightsquigarrow \mathcal{S}, \quad \mathcal{R} \subseteq \mathcal{S} \end{array} \right. \Rightarrow \mathcal{F}(\mathcal{R}) \rightsquigarrow \mathcal{F}(\mathcal{S})$$

The main difference w.r.t. Definition 1.5 is in clause (1): instead of respecting silent evolutions, a weakly monotonic function has to respect silent simulations. On the visible side (2), we suppose that  $\mathcal{R}$  and  $\mathcal{S}$  are silent simulations. The immediate consequence of these modifications appears in the following result: the up-to function may only be used on visible evolutions, and the candidate relation  $\mathcal{R}$  has to be a silent simulation.

**Proposition 2.2 (correctness of weakly monotonic functions)** *Let  $\mathcal{F}$  be weakly monotonic. If  $\mathcal{R} \rightsquigarrow \mathcal{R}$ , and  $\mathcal{R} \rightsquigarrow \mathcal{F}(\mathcal{R})$ , then  $\mathcal{F}^*(\mathcal{R})$  is a simulation.*

*Proof.*  $\mathcal{R} \rightsquigarrow \mathcal{R}$  and (1) in the weak monotonicity of  $\mathcal{F}$  give  $\mathcal{F}^n(\mathcal{R}) \rightsquigarrow \mathcal{F}^n(\mathcal{R})$  for all  $n$ , by induction on  $n$ . Then, by a second induction on  $n$ , we get for all  $n$ ,  $\mathcal{F}^n(\mathcal{R}) \rightsquigarrow \mathcal{F}^{n+1}(\mathcal{R})$ . We conclude with Lemma 1.3(3).  $\square$

Now we study the class of weakly monotonic functions: the following lemma ensures that the functions given by Lemma 1.7 can be used in the setting of weakly monotonic functions. Furthermore, weakly monotonic functions can be composed using the most important constructors:

**Lemma 2.3** *Any monotonic function is weakly monotonic; composition ( $\circ$ ), union ( $\cup$ ), iteration ( $*$ ) and chaining ( $\frown$ ) respect weak monotonicity.*

*Proof.* We prove the respectfulness of chaining. Let  $\mathcal{F}$  and  $\mathcal{G}$  be two weakly monotonic functions.

1. Suppose  $\mathcal{R} \rightsquigarrow \mathcal{R}$ , then  $\mathcal{F}(\mathcal{R}) \rightsquigarrow \mathcal{F}(\mathcal{R})$  and  $\mathcal{G}(\mathcal{R}) \rightsquigarrow \mathcal{G}(\mathcal{R})$  by weak monotonicity of  $\mathcal{F}$  and  $\mathcal{G}$ . We get the leftmost diagram below, that we can close using Lemma 1.3(1) (rightmost diagram).

$$\begin{array}{cccccc} P & \mathcal{F}(\mathcal{R}) & P_1 & \mathcal{G}(\mathcal{R}) & Q & & P & \mathcal{F}(\mathcal{R}) & P_1 & \mathcal{G}(\mathcal{R}) & Q \\ \tau \downarrow & & \tau \downarrow & & & & \tau \downarrow & & \tau \downarrow & & \downarrow \tau \\ P' & \mathcal{F}(\mathcal{R}) & P'_1 & & & & P' & \mathcal{F}(\mathcal{R}) & P'_1 & \mathcal{G}(\mathcal{R}) & Q' \end{array}$$

2. Suppose  $\mathcal{R} \rightsquigarrow \mathcal{R}$ ,  $\mathcal{R} \rightsquigarrow \mathcal{S}$ ,  $\mathcal{S} \rightsquigarrow \mathcal{S}$  and  $\mathcal{R} \subseteq \mathcal{S}$ . By weak monotonicity of  $\mathcal{F}$ , we have  $\mathcal{F}(\mathcal{R}) \rightsquigarrow \mathcal{F}(\mathcal{S})$ ;  $\mathcal{G}$  is weakly monotonic, so that  $\mathcal{G}(\mathcal{R}) \rightsquigarrow \mathcal{G}(\mathcal{S})$ ,  $\mathcal{G}(\mathcal{R}) \rightsquigarrow \mathcal{G}(\mathcal{R})$  and  $\mathcal{G}(\mathcal{S}) \rightsquigarrow \mathcal{G}(\mathcal{S})$ . With Lemma 1.3(2) and simple diagram chasing arguments, we prove  $\mathcal{F}(\mathcal{R})\mathcal{G}(\mathcal{R}) \rightsquigarrow \mathcal{F}(\mathcal{S})\mathcal{G}(\mathcal{S})$ .  $\square$

The closure under the chaining constructor naturally suggests the use of interesting up-to techniques, and in particular *up to transitivity*, given by  $\mathcal{F}(\mathcal{R}) = \mathcal{R}^*$ , and *up to weak bisimilarity*, using  $\mathcal{F}(\mathcal{R}) = \approx \mathcal{R} \approx$ .

## 2.2 Combining Monotonicity and Weak Monotonicity

In introducing weakly monotonic functions, we have restricted the use of up-to techniques to visible steps. We show how to develop further this approach by combining a monotonic function and a weakly monotonic function so as to employ constrained up-to techniques on silent steps, and full-fledged up-to techniques on visible steps.

**Proposition 2.4 (unified up-to technique)** *Let  $\mathcal{F}$  be monotonic and  $\mathcal{G}$  be weakly monotonic, and suppose further that  $\mathcal{F} \subseteq \mathcal{G}$ .*

*If  $\mathcal{R} \succ_{\mathcal{F}} \mathcal{F}(\mathcal{R})$  and  $\mathcal{R} \succ_{\mathcal{G}} \mathcal{G}(\mathcal{R})$ , then  $(\mathcal{G}^*)^*(\mathcal{R})$  is a simulation.*

*Proof.* First, we prove  $\mathcal{F}^* \subseteq \mathcal{G}^*$  (1) and  $(\mathcal{G}^*)^*(\mathcal{F}^*(\mathcal{R})) = (\mathcal{G}^*)^*(\mathcal{R})$  (2). Then we show  $\forall n, \mathcal{F}^n(\mathcal{R}) \succ_{\mathcal{F}} \mathcal{F}^{n+1}(\mathcal{R})$  by induction on  $n$ , using the fact that  $\mathcal{R} \succ_{\mathcal{F}} \mathcal{F}(\mathcal{R})$  and the monotonicity of  $\mathcal{F}$ . This leads to  $\mathcal{F}^*(\mathcal{R}) \succ_{\mathcal{F}} \mathcal{F}^*(\mathcal{R})$  (3).

We obtain  $\mathcal{F}^*(\mathcal{R}) \succ_{\mathcal{G}} \mathcal{G}^*(\mathcal{R})$  (4) by proving  $\mathcal{F}^n(\mathcal{R}) \succ \mathcal{G}^{n+1}(\mathcal{R})$ , by induction on  $n$ :

- $n = 0$ : the hypotheses  $\mathcal{F} \subseteq \mathcal{G}$ ,  $\mathcal{R} \succ_{\mathcal{F}} \mathcal{F}(\mathcal{R})$  and  $\mathcal{R} \succ_{\mathcal{G}} \mathcal{G}(\mathcal{R})$  entail that  $\mathcal{R} \succ \mathcal{G}(\mathcal{R})$ .
- $n > 0$ : the inductive hypothesis is  $\mathcal{F}^{n-1}(\mathcal{R}) \succ \mathcal{G}^n(\mathcal{R})$ . Using (1) and the monotonicity of  $\mathcal{F}$ , we get  $\mathcal{F}^n(\mathcal{R}) \succ \mathcal{F}(\mathcal{G}^n(\mathcal{R})) \subseteq \mathcal{G}^{n+1}(\mathcal{R})$ .

$\mathcal{G}^*(\mathcal{R}) = \mathcal{G}^*(\mathcal{F}^*(\mathcal{R}))$ , so that (4) leads to  $\mathcal{F}^*(\mathcal{R}) \succ_{\mathcal{G}} \mathcal{G}^*(\mathcal{F}^*(\mathcal{R}))$  (5). Since  $\mathcal{G}$  is weakly monotonic,  $\mathcal{G}^*$  also from Lemma 2.3. Then, Proposition 2.2, applied with (3) and (5) to the candidate relation  $\mathcal{F}^*(\mathcal{R})$  ensures that  $(\mathcal{G}^*)^*(\mathcal{F}^*(\mathcal{R}))$  is a simulation. The result finally comes from (2).  $\square$

The following theorem is the counterpart of Theorem 1.9 in the richer setting we have introduced:

**Theorem 2.5** *If  $\left\{ \begin{array}{l} \mathcal{R} \succ_{\mathcal{F}} \succ \mathcal{R}^{\approx} \approx \\ \mathcal{R} \succ_{\mathcal{G}} (\mathcal{R} \cup \approx)^* \end{array} \right.$  and  $\left\{ \begin{array}{l} \mathcal{R}^{-1} \succ_{\mathcal{F}} \succ \mathcal{R}^{-1} \approx \approx \\ \mathcal{R}^{-1} \succ_{\mathcal{G}} (\mathcal{R}^{-1} \cup \approx)^* \end{array} \right.$  then  $\mathcal{R} \subseteq \approx$ .*

*Proof.*  $\mathcal{F}(\mathcal{R}) \triangleq \succ \mathcal{R}^{\approx} \approx$  and  $\mathcal{G}(\mathcal{R}) \triangleq (\mathcal{R} \cup \approx)^*$  satisfy the conditions of Proposition 2.4 ( $\succ \subseteq \approx$ ), so that  $\mathcal{G}(\mathcal{R})$  and  $\mathcal{G}(\mathcal{R}^{-1})$  are simulations ( $\mathcal{G} = (\mathcal{G}^*)^*$ ). Since  $\mathcal{G}(\mathcal{R})^{-1} = \mathcal{G}(\mathcal{R}^{-1})$ ,  $\mathcal{G}(\mathcal{R})$  is a bisimulation, and  $\mathcal{R} \subseteq \mathcal{G}(\mathcal{R}) \subseteq \approx$ .  $\square$

We thus have a modular theory of up-to techniques for weak bisimulation that follows the approach for the strong case in [7]. Technically, the main improvement over previous works is the ability to exploit weaker hypotheses when reasoning about visible steps: for instance, *up to transitivity* ( $\mathcal{R} \succ_{\mathcal{F}} \mathcal{R}^*$ ) and *up to weak bisimilarity* ( $\mathcal{R} \succ_{\mathcal{G}} \approx \mathcal{R} \approx$ ) techniques entail valid proof methods.

### 2.3 ‘Up to Context’ Proof Techniques

An important family of up-to techniques that has not been discussed yet is ‘up to context’. When the states of the LTS are described by a syntax, such techniques make it possible to remove common sub-terms of the processes being compared along the bisimulation game, and thus help reducing the size of the relation one has to exhibit. Up to context proof techniques have been mostly used in the context of CCS and the  $\pi$ -calculus [7]. We now show how reasoning up to context can be achieved in our setting.

We denote by  $\tilde{P}$  a vector of processes, and by  $P^i$  the  $i$ -th component of such vector. We call (*polyadic*) *context of arity  $n$*  a function from vectors of size  $n$  to processes (we adopt an approach that allows us to abstract over the details of the underlying syntax). We let  $C, D$  range over contexts, and denote by  $C[\tilde{P}]$  the application of a context  $C$  to a vector of processes  $\tilde{P}$  (we shall implicitly assume that the size of  $\tilde{P}$  and the arity of  $C$  are equal). We let  $\mathcal{C}, \mathcal{D}$  range over families of contexts. Given a family  $\mathcal{C}$  of contexts, we define the closure up to  $\mathcal{C}$  function by  $\mathcal{C}(\mathcal{R}) \triangleq \{ (C[\tilde{P}], C[\tilde{Q}]), C \in \mathcal{C} \text{ and } \forall i, P^i \mathcal{R} Q^i \}$ .

In the following technical definition, we use notations  $\xrightarrow{\epsilon}$  and  $\xrightarrow{\epsilon\delta}$  as synonyms for the identity relation  $\mathcal{I}$  (we suppose  $\epsilon \notin \mathcal{L}$ ). Furthermore, we write  $\tilde{P} \xrightarrow{\tilde{\delta}} \tilde{P}'$  (resp.  $\tilde{P} \xrightarrow{\tilde{\delta}} \tilde{P}'$ ) for  $\forall i, P^i \xrightarrow{\delta^i} P'^i$  (resp.  $\forall i, P^i \xrightarrow{\delta^i} P'^i$ ).

**Definition 2.6 (faithfulness)** *Let  $\mathcal{C}$  be a family of contexts. We say that  $\mathcal{C}$  is faithful if for all  $C \in \mathcal{C}$ , whenever  $C[\tilde{P}] \xrightarrow{\alpha} R$ , there are  $C' \in \mathcal{C}$ ,  $\tilde{P}'$ , and a vector  $\tilde{\delta}$  whose components are in  $\mathcal{L} \cup \{\epsilon\}$  such that:*

1.  $R = C'[\tilde{P}']$  and  $\tilde{P} \xrightarrow{\tilde{\delta}} \tilde{P}'$ ;
2. for all  $\tilde{Q}, \tilde{Q}'$  such that  $\tilde{Q} \xrightarrow{\tilde{\delta}} \tilde{Q}'$  and  $C[\tilde{Q}] \xrightarrow{\alpha} C'[\tilde{Q}']$ ;
3. if  $\alpha = \tau$  then the components of  $\tilde{\delta}$  are taken in  $\{\tau\} \cup \{\epsilon\}$ .

A context  $C$  is faithful if it belongs to a faithful family of contexts.

This is the direct adaptation to the weak case of the notion of faithfulness in [7], to which we add the restriction (3) for silent evolutions. We return to this additional clause below.

**Proposition 2.7** *The closure up to a faithful family of contexts is monotonic.*

*Proof.* We consider the case of contexts of arity 1, and prove separately the two implications of Definition 1.5.

- Suppose  $\mathcal{R} \xrightarrow{\tau} \mathcal{S}$ ,  $\mathcal{R} \subseteq \mathcal{S}$ ,  $P \mathcal{C}(\mathcal{R}) Q$ , with  $P = C[P_0]$ ,  $Q = C[Q_0]$ ,  $P_0 \mathcal{R} Q_0$  and  $P \xrightarrow{\tau} P'$ . By faithfulness, there is  $C', P'_0$  and  $\delta$  such that  $P' = C'[P'_0]$  and  $P_0 \xrightarrow{\delta} P'_0$ . From (3),  $\delta$  is either  $\tau$  or  $\epsilon$ :
  - $\delta = \epsilon$ : using (2),  $C[Q_0] \xrightarrow{\tau} C'[Q_0]$  and since  $\mathcal{R} \subseteq \mathcal{S}$ , we have  $P_0 \mathcal{S} Q_0$ .
  - $\delta = \tau$ : since  $P_0 \mathcal{R} Q_0$  and  $\mathcal{R} \xrightarrow{\tau} \mathcal{S}$  there is  $Q'_0$  such that  $Q_0 \xrightarrow{\tau} Q'_0$  and  $P'_0 \mathcal{S} Q'_0$ . Using (2), we get  $C[Q_0] \xrightarrow{\tau} C'[Q'_0]$ .
- Suppose  $\mathcal{R} \xrightarrow{\alpha} \mathcal{S}$ ,  $\mathcal{R} \subseteq \mathcal{S}$ ,  $P \mathcal{C}(\mathcal{R}) Q$ , with  $P = C[P_0]$ ,  $Q = C[Q_0]$ ,  $P_0 \mathcal{R} Q_0$  and  $P \xrightarrow{\alpha} P'$ . By faithfulness, there is  $C', P'_0$  and  $\delta$  such that  $P' = C'[P'_0]$  and  $P_0 \xrightarrow{\delta} P'_0$ .

- $\delta = \epsilon$  is similar to the previous case
- $\delta = \alpha \in \mathcal{L}$ : since  $P_0 \mathcal{R} Q_0$  and  $\mathcal{R} \rightsquigarrow \mathcal{S}$  there is  $Q'_0$  such that  $Q_0 \xrightarrow{\alpha} Q'_0$  and  $P'_0 \mathcal{S} Q'_0$ . Using (2), we get  $C[Q_0] \xrightarrow{\alpha} C'[Q'_0]$ .  $\square$

This result ensures that we can use Theorem 2.5 to reason up to faithful contexts both on visible and silent steps. In the strong up-to theory of [7], all CCS contexts are faithful, as well as all *non input-guarded*  $\pi$  contexts (i.e. those where the argument process is not placed under an input prefix). This is not the case in our setting: closure by parallel composition (given by  $C_Q[P] = P \mid Q$ ) is not faithful. This is due to the restriction (3): when  $C[P] \xrightarrow{\tau} R$  we require that the context either does the silent action itself ( $C[P] = \tau.P$ ), or delegates it to  $P$  ( $C[P] = P$ ), but the silent action cannot follow from an interaction between the context and a visible action of  $P$  (in CCS, e.g.,  $C[P] = \bar{a} \mid P$  and  $P = a.P'$ ). This restriction is a consequence of the separation between silent and visible actions in Definition 1.5: in order to prove  $\mathcal{C}(\mathcal{R}) \xrightarrow{\tau} \mathcal{C}(\mathcal{S})$ , we only suppose  $\mathcal{R} \xrightarrow{\tau} \mathcal{S}$  (while working in the setting of [7] would mean supposing  $\mathcal{R} \rightsquigarrow \mathcal{S}$ ). Therefore, when we observe the silent evolution of a process  $C[P]$ , we have no hypothesis to reason about the case where  $P$  does a visible evolution. Formulating our results with the original definition of faithfulness would have been possible up to Proposition 2.4 and Theorem 2.5, that inherently exploit a separation between visible and silent transitions, and thus render clause (3) necessary in Definition 2.6.

To comment further on this restriction, let us remark that one of the main motivations of this work is to provide useful proof techniques to reason about rather large systems, such as in [4]. In such a setting, it is likely that we face a complex LTS, where visible and silent transitions, while of course being correlated in the behaviour of the system, belong to ‘orthogonal components’ of the LTS. In such situations, it seems likely that clause (3) does not prevent the use of up to context proof techniques.

## 3 Beyond Expansion

### 3.1 Controlled Relations

In this section, we enrich our framework with the possibility to use alternatives to  $\succsim$  (which is the best we can do using Theorem 2.5) to handle  $\tau$  transitions in bisimulation proofs. We define a class of relations that are controlled w.r.t. silent transitions, meaning that they prevent silent steps from being cancelled in an up-to bisimulation game.

The left-chaining functions associated to such relations are not weakly monotonic, and we thus have to depart from the theory we have developed so far. Roughly, a controlled relation is defined as a relation that induces a correct proof technique when used as a left-chaining up-to technique. The following technical definition defines a uniform way to plug a non weakly monotonic left-chaining function into our setting.

**Definition 3.1 (controlled relation)**  $\mathcal{B}$  is a controlled relation if the following holds for all relations  $\mathcal{R}, \mathcal{S}$ :

$$(1) \mathcal{R} \succ_{\mathcal{B}} \mathcal{B}^* \mathcal{R} \Rightarrow \mathcal{B}^* \mathcal{R} \succ_{\mathcal{B}} \mathcal{B}^* \mathcal{R} \quad (2) \begin{cases} \mathcal{R} \succ_{\mathcal{B}} \mathcal{B}^* \mathcal{R} \\ \mathcal{R} \succ_{\mathcal{B}} \mathcal{S}, \quad \mathcal{S} \succ_{\mathcal{B}} \mathcal{S} \end{cases} \Rightarrow \mathcal{B}^* \mathcal{R} \succ_{\mathcal{B}} \mathcal{B}^* \mathcal{S}.$$

**Remark 3.2** *Technically, a controlled relation need not be a simulation. However, by taking  $\mathcal{R} = \mathcal{S} = \mathcal{I}$ , we see that if  $\mathcal{B}$  is a controlled relation, then  $\mathcal{B}^*$  is a simulation. Furthermore, the union of two controlled relations is not necessarily a controlled relation. Thus, this does not a priori induce a notion of controlled bisimilarity. We say that  $\mathcal{B}$  is a controlled bisimulation if it is a controlled relation contained in bisimilarity.*

**Definition 3.3 (transparency)** *Given a relation  $\mathcal{B}$  and a function  $\mathcal{F}$ ,  $\mathcal{F}$  is  $\mathcal{B}$ -transparent if  $\mathcal{F}(\mathcal{B}^* \mathcal{R}) \subseteq \mathcal{B}^* \mathcal{F}(\mathcal{R})$  for any relation  $\mathcal{R}$ .*

*$\mathcal{F}$  is transparent if it is  $\mathcal{B}$ -transparent for any relation  $\mathcal{B}$ .*

This transparency property is necessary to compute fixpoints in the proof of the following proposition.

**Proposition 3.4 (up to controlled relation)** *Let  $\mathcal{F}$  and  $\mathcal{G}$  be two functions, and  $\mathcal{B}$  a relation such that:  $\mathcal{B}$  is a controlled relation,  $\mathcal{F}$  is monotonic and  $\mathcal{B}$ -transparent,  $\mathcal{G}$  is weakly monotonic. Suppose moreover that  $\mathcal{G}$  contains  $\mathcal{F}$  and  $\mathcal{B}^* \bullet$ . If  $\mathcal{R} \succ_{\mathcal{B}} \mathcal{B}^* \mathcal{F}(\mathcal{R})$  and  $\mathcal{R} \succ_{\mathcal{B}} \mathcal{G}(\mathcal{R})$ , then  $(\mathcal{G}^*)^*(\mathcal{R})$  is a simulation.*

*Proof.* We reason along the lines of the proof of Proposition 2.4. First, we prove  $\mathcal{F}^*(\mathcal{R}) \succ_{\mathcal{B}} \mathcal{B}^* \mathcal{F}^*(\mathcal{R})$ , using the monotonicity of  $\mathcal{F}$ ,  $\mathcal{R} \succ_{\mathcal{B}} \mathcal{B}^* \mathcal{F}(\mathcal{R})$  and the  $\mathcal{B}$ -transparency of  $\mathcal{F}$ . Since  $\mathcal{B}$  is controlled, we get  $\mathcal{B}^* \mathcal{F}^*(\mathcal{R}) \succ_{\mathcal{B}} \mathcal{B}^* \mathcal{F}^*(\mathcal{R})$  (1).  $\forall n, \mathcal{B}^* \mathcal{F}^n(\mathcal{R}) \subseteq \mathcal{B}^* \mathcal{G}^n(\mathcal{R}) \subseteq \mathcal{G}^{n+1}(\mathcal{R})$  (2) follows by induction on  $n$ , using  $\mathcal{F} \subseteq \mathcal{G}$  and  $\mathcal{B}^* \bullet \subseteq \mathcal{G}$ . Then we show  $\forall n, \mathcal{F}^n(\mathcal{R}) \succ_{\mathcal{B}} \mathcal{G}^{n+2}(\mathcal{R})$  (3) by induction on  $n$ :

- $n = 0$ : the visible case is immediate; for the silent case, we have  $\mathcal{R} \succ_{\mathcal{B}} \mathcal{B}^* \mathcal{F}(\mathcal{R}) \subseteq \mathcal{B}^* \mathcal{G}(\mathcal{R}) \subseteq \mathcal{G}^2(\mathcal{R})$ .
- $n > 0$ : the inductive hypothesis is  $\mathcal{F}^{n-1}(\mathcal{R}) \succ_{\mathcal{B}} \mathcal{G}^{n+1}(\mathcal{R})$ . Using (2) and the monotonicity of  $\mathcal{F}$ , we obtain  $\mathcal{F}^n(\mathcal{R}) \succ_{\mathcal{B}} \mathcal{F}(\mathcal{G}^{n+1}(\mathcal{R})) \subseteq \mathcal{G}^{n+2}(\mathcal{R})$ .

Using (2),  $\mathcal{G}^*(\mathcal{R}) \subseteq \mathcal{G}^*(\mathcal{B}^* \mathcal{F}^*(\mathcal{R}))$ , so that (3) gives  $\mathcal{F}^*(\mathcal{R}) \succ_{\mathcal{B}} \mathcal{G}^*(\mathcal{B}^* \mathcal{F}^*(\mathcal{R}))$  (4). With the weak monotonicity of  $\mathcal{G}$ , we get  $\mathcal{G}^*(\mathcal{B}^* \mathcal{F}^*(\mathcal{R})) \succ_{\mathcal{B}} \mathcal{G}^*(\mathcal{B}^* \mathcal{F}^*(\mathcal{R}))$  (5). We can then use the fact that  $\mathcal{B}$  is a controlled relation with (1), (4) and (5), so that we obtain  $\mathcal{B}^* \mathcal{F}^*(\mathcal{R}) \succ_{\mathcal{B}} \mathcal{B}^* \mathcal{G}^*(\mathcal{B}^* \mathcal{F}^*(\mathcal{R}))$ , which leads, using  $\mathcal{B}^* \mathcal{G}^* \subseteq \mathcal{G}^*$ , to  $\mathcal{B}^* \mathcal{F}^*(\mathcal{R}) \succ_{\mathcal{B}} \mathcal{G}^*(\mathcal{B}^* \mathcal{F}^*(\mathcal{R}))$  (6). Proposition 2.2, applied with (1) and (6) to  $\mathcal{G}^*$  and  $\mathcal{B}^* \mathcal{F}^*(\mathcal{R})$ , ensures that  $(\mathcal{G}^*)^*(\mathcal{B}^* \mathcal{F}^*(\mathcal{R}))$  is a simulation. The result finally follows from  $(\mathcal{G}^*)^*(\mathcal{R}) = (\mathcal{G}^*)^*(\mathcal{B}^* \mathcal{F}^*(\mathcal{R}))$ .  $\square$

**Remark 3.5** *While in practise, most of our functions will satisfy  $\mathcal{G}^{**} = \mathcal{G}^*$  or even  $\mathcal{G}^* = \mathcal{G}$ , we can imagine monotonic functions such that this is not true. Indeed, consider the following function<sup>1</sup>  $f$ , defined over sets of integers:*

$$X \mapsto \begin{cases} \mathbb{N} & \text{if } \mathbb{N}^* \subseteq X, \\ X \cup \{\min(\mathbb{N}^*/X)\} & \text{otherwise.} \end{cases}$$

*It is monotonic (w.r.t.  $\subseteq$ ), but  $f^*(\{1\}) = \mathbb{N}^* \neq \mathbb{N} = f^{**}(\{1\})$ .*

<sup>1</sup>We thank Emmanuel Jeandel for this counter-example.

**Lemma 3.6** *The identity and all  $\mathcal{S}$ -right-chaining functions are transparent. If  $\mathcal{B} \subseteq \mathcal{S}$  then the constant-to- $\mathcal{S}$  function is  $\mathcal{B}$ -transparent. Given a family of contexts  $\mathcal{C}$ , if  $\mathcal{C}(\mathcal{B}) \subseteq \mathcal{B}$  (i.e. “ $\mathcal{B}$  is a  $\mathcal{C}$ -congruence”), then the closure up to  $\mathcal{C}$  function is  $\mathcal{B}$ -transparent.*

*The composition, union and iteration constructors respect  $\mathcal{B}$ -transparency.*

In practise, we will work with  $\mathcal{S} = \approx$  and require that  $\mathcal{B} \subseteq \approx$ , so that condition  $\mathcal{B} \subseteq \mathcal{S}$  will be satisfied.

The chaining constructor does not respect  $\mathcal{B}$ -transparency, but this would be of little use anyway: Proposition 3.4 indeed requires the transparency of a monotonic function, which rules out the chaining constructor, that does not respect monotonicity.

Also notice that  $\succsim \bullet$ , the expansion-left-chaining function, is not transparent in general. This hence prevents us from encompassing the up to expansion proof technique in the statement of the following theorem.

**Theorem 3.7** *Let  $\mathcal{B}$  be a controlled bisimulation.*

$$\text{If } \begin{cases} \mathcal{R} \succsim \mathcal{B}^* \mathcal{R} \approx \\ \mathcal{R} \succsim (\mathcal{R} \cup \approx)^* \end{cases} \text{ and } \begin{cases} \mathcal{R}^{-1} \succsim \mathcal{B}^* \mathcal{R}^{-1} \approx \\ \mathcal{R}^{-1} \succsim (\mathcal{R}^{-1} \cup \approx)^* \end{cases} \text{ then } \mathcal{R} \subseteq \approx.$$

*Proof.*  $\mathcal{B}$ ,  $\mathcal{F}(\mathcal{R}) \triangleq \mathcal{R} \approx$  and  $\mathcal{G}(\mathcal{R}) \triangleq (\mathcal{R} \cup \approx)^*$  satisfy the conditions of Proposition 3.4, so that  $\mathcal{G}(\mathcal{R})$  and  $\mathcal{G}(\mathcal{R}^{-1})$  are simulations ( $\mathcal{G} = (\mathcal{G}^*)^*$ ). Since  $\mathcal{G}(\mathcal{R})^{-1} = \mathcal{G}(\mathcal{R}^{-1})$ ,  $\mathcal{G}(\mathcal{R})$  is a bisimulation, and  $\mathcal{R} \subseteq \mathcal{G}(\mathcal{R}) \subseteq \approx$ .  $\square$

This theorem is the counterpart of Theorem 2.5 using a controlled bisimulation instead of  $\succsim$ . A refined version of this result, in which two distinct controlled bisimulations are used for the silent evolutions of  $\mathcal{R}$  and  $\mathcal{R}^{-1}$ , also holds. This can be useful in particular because the class of controlled bisimulations is not closed under union, as explained by Remark 3.2.

Moreover, if we need the closure under a family of contexts:  $\mathcal{R} \succsim \mathcal{B}^* \mathcal{C}(\mathcal{R}) \approx$  and  $\mathcal{R} \succsim (\mathcal{C}(\mathcal{R}) \cup \approx)^*$ , we have to ensure that the controlled bisimulation  $\mathcal{B}$  is a  $\mathcal{C}$ -congruence (see Lemma 3.6), which can be quite difficult. In such cases, one can restrict the family of contexts used on silent actions to  $\mathcal{D} \subseteq \mathcal{C}$ , with  $\mathcal{R} \succsim \mathcal{B}^* \mathcal{D}(\mathcal{R}) \approx$ , in order to weaken the congruence condition.

The following lemma gives a way to prove that a controlled relation is a controlled bisimulation. The remainder of the section is devoted to the construction of controlled relations.

**Lemma 3.8** *If  $\mathcal{B}$  is a controlled relation and  $\mathcal{B}^{-1} \succsim \mathcal{B}^{-1} \cup \approx$ , then  $\mathcal{B}$  is a controlled bisimulation.*

*Proof.* With Lemma 1.3(3),  $\mathcal{B}^{-1} \cup \approx$  is a simulation; from Remark 3.2,  $\mathcal{B}^*$  as well, so that  $\mathcal{B}^* \cup \approx$  is a bisimulation.  $\square$

## 3.2 Relaxed Expansion

**Definition 3.9 (relaxed expansion)** *A relation  $\mathcal{E}$  is a relaxed expansion if whenever  $P \mathcal{E} Q$ ,*

1.  $P \succsim P'$  implies  $Q \succsim Q'$  and  $P' \mathcal{E} Q'$  for some  $Q'$  or  $P' \mathcal{E} Q$ ,

2.  $P \xrightarrow{a} P'$  implies  $Q \xrightarrow{a} Q'$  and  $P' \mathcal{E} Q'$  for some  $Q'$ .

Relaxed expansion, denoted by  $\succsim$ , is the union of all relaxed expansions  $\mathcal{E}$  such that  $\mathcal{E}^{-1}$  is a simulation.

When  $P \succsim Q$  and  $P \xrightarrow{a} P'$ ,  $Q$  has to do immediately an  $a$  transition, but then can do as many silent transitions as necessary. The intuition behind the definition of relaxed expansion is that, using this possibility,  $Q$  can do some ‘preliminary internal computation’ in order to be able to remain faster than  $P$  until the next visible action.

**Lemma 3.10**  $\succsim$  is a relaxed expansion, and the following strict inclusions hold:

$$\succ \subsetneq \succsim \subsetneq \approx.$$

*Proof.* The first point and the inclusions are straightforward. We illustrate the strictness of the inclusions using CCS processes:  $a.b \succsim a.\tau.b$  holds but not  $a.b \succ a.\tau.b$ , and  $a \approx \tau.a$  holds but not  $a \succsim \tau.a$ .  $\square$

**Theorem 3.11** A relaxed expansion is a controlled relation.  $\succsim$  is a controlled bisimulation.

*Proof.* We show that if  $\mathcal{E}$  is a relaxed expansion, then it is also the case for  $\mathcal{E}^*$ . Both results follow easily.  $\square$

In general,  $\succsim$  is not a congruence: for instance, in CCS,  $a.b \succsim a.\tau.b$  holds but not  $\bar{a} \mid a.b \succsim \bar{a} \mid a.\tau.b$ . This is somewhat related to the problem of up to parallel composition, discussed in Subsection 2.3: as contexts may turn a visible action into a silent one, the way we put a stress on visible transitions in the definition of  $\succsim$  somehow vanishes when adding parallel components.

[8] defines *almost weak bisimilarity*. This relation is very close to  $\succsim$ , but coarser; it is a controlled bisimulation, and it is not a congruence in general. We preferred our version because it fits better within the style of the definitions of behavioural equivalences in our presentation.

### 3.3 Introducing Termination Guarantees

We now show how to obtain controlled relations using termination guarantees. The theorems below follow from general results about commuting diagrams, presented in Sect. 4. Their proofs are thus deferred to that section.

**Theorem 3.12** Let  $\mathcal{B}$  be a relation such that  $\mathcal{B} \mapsto \mathcal{B}^+$  and  $\mathcal{B}$  terminates. Then  $\mathcal{B}$  is a controlled relation.

**Theorem 3.13** Let  $\mathcal{B}$  be a relation such that  $\mathcal{B} \mapsto \mathcal{B}^*$  and  $\mathcal{B}^+ \xrightarrow{\tau}^+ \mathcal{B}^*$  terminates. Then  $\mathcal{B}$  is a controlled relation.

Unlike  $\succsim$ , where the control on silent moves is fixed by the co-inductive definition of the relation, in these two results we start with a relation that roughly respects the – too permissive – weak bisimulation game, and constrain it a posteriori, in such a way that it cannot cancel silent steps indefinitely.

For example, the erroneous up-to relation  $\mathcal{B} = \{(a, \tau.a)\}$  is rejected because  $\mathcal{B}$  evolves to  $\mathcal{I} = \mathcal{B}^0$ , and  $\mathcal{B}^{+\tau^+} = \{(a, a)\}$  obviously does not terminate.

In contrast with the theory developed so far, there is no direct way to define the greatest relation satisfying the requirements in Theorems 3.12 and 3.13, the main reason being that the union of terminating relations does not terminate in general. Also remark that the termination of  $\mathcal{B}^{+\tau^+}$  does not entail the termination of  $\mathcal{B}$  or  $\tau$ . Theorem 3.13 can thus be applied to systems exhibiting infinite chains of  $\tau$  transitions ( $\pi$  or CCS with replication...).

There are processes that are not related by  $\approx$ , but by a relation satisfying the conditions of the previous theorems: consider  $(a + a, \tau.a)$  or  $(a \mid (\nu b)b, \tau.a)$ . Sect. 5 presents a more refined example of such a situation.

We can use the up-to techniques we have defined previously to show the evolution condition in the above theorems ( $\mathcal{B} \rightsquigarrow \mathcal{B}^+$  or  $\mathcal{B} \rightsquigarrow \mathcal{B}^*$ ). However one has to be careful, because the simulation relation obtained with these techniques is  $\mathcal{F}^*(\mathcal{B})$ . Depending on  $\mathcal{F}$ , this relation may be reflexive, which discards Theorem 3.12, or just quite complex, so that proving the termination of  $\mathcal{F}^*(\mathcal{B})$  or  $\mathcal{F}^*(\mathcal{B})^{+\tau^+}$  may be delicate.

**Corollary 3.14** *Let  $\mathcal{R}$  be a relation.*

*If  $\mathcal{R}^{-1} \rightsquigarrow \mathcal{R}^{-1} \cup \approx$  and  $\left\{ \begin{array}{l} \mathcal{R} \rightsquigarrow \mathcal{R}^* \text{ and } \mathcal{R}^{+\tau^+} \text{ terminates} \\ \text{or } \mathcal{R} \rightsquigarrow \mathcal{R}^+ \text{ and } \mathcal{R} \text{ terminates,} \end{array} \right.$  then  $\mathcal{R} \subseteq \approx$ .*

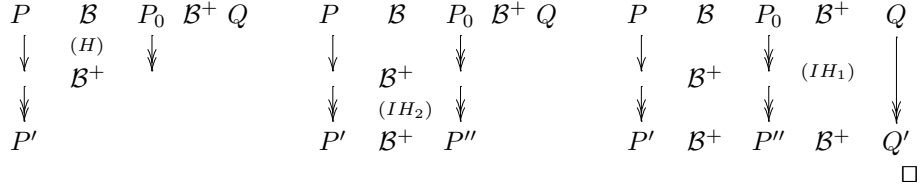
## 4 Results about Commuting Diagrams

In this section, we work in the more general setting of *diagrams*, commonly found in rewriting theory. In addition to  $\mathcal{R}, \mathcal{S}$  we let  $\rightarrow, \leftrightarrow$  and  $\rightsquigarrow$  range over relations. As before,  $\rightarrow^+$  (resp.  $\rightsquigarrow^+$ ) is the transitive (resp. reflexive transitive) closure of  $\rightarrow$ . We shall say that four relations  $(\mathcal{R}, \rightarrow, \mathcal{S}, \leftrightarrow)$  form a *diagram*, denoted  $(\mathcal{R}, \rightarrow) \gg (\mathcal{S}, \leftrightarrow)$ , if whenever  $P \mathcal{R} Q$  and  $P \rightarrow P'$ , there is  $Q'$  such that  $P' \mathcal{S} Q'$  and  $Q \leftrightarrow Q'$  (in our proofs, we shall sometimes adopt the usual graphical notation for diagrams). We say that two relations  $\mathcal{R}$  and  $\rightarrow$  *commute* if  $(\mathcal{R}, \rightarrow) \gg (\mathcal{R}, \rightarrow)$ . Notice that a relation  $\mathcal{R}$  is a simulation iff  $\mathcal{R}$  commutes with  $\xrightarrow{\alpha}$  for all  $\alpha \in \mathcal{L}$ .

### 4.1 A First Termination Argument

**Lemma 4.1** *Let  $\mathcal{B}, \rightarrow$  be two relations such that  $\mathcal{B}$  terminates. If  $(\mathcal{B}, \rightarrow) \gg (\mathcal{B}^+, \rightsquigarrow)$ , then  $\mathcal{B}^+$  and  $\rightsquigarrow$  commute.*

*Proof.* Let  $\phi(P')$  be the following predicate over processes: “For all  $P, Q$  such that  $P \rightsquigarrow P'$  and  $P \mathcal{B}^+ Q$ , there is  $Q'$  such that  $Q \rightsquigarrow Q'$  and  $P' \mathcal{B}^+ Q'$ ”. We prove that  $\phi$  is true for any process by induction over the well-founded relation  $\mathcal{B}^{-1}$ , which leads to the induction hypothesis ( $IH_1$ ):  $\forall P'', P' \mathcal{B}^+ P'' \Rightarrow \phi(P'')$ . Then we do a second induction on the derivation  $P \rightsquigarrow P'$ , leading to a second induction hypothesis: ( $IH_2$ ). The interesting case is represented on the first following diagram, where we close the first step using the hypothesis. We use the internal induction to obtain the second diagram, and the main induction to close the whole diagram (we check that  $P' \mathcal{B}^+ P''$ ).



**Remark 4.2** *The commutation hypothesis  $(\mathcal{B}, \rightarrow) \gg (\mathcal{B}^+, \rightarrow)$  cannot be weakened to  $(\mathcal{B}, \rightarrow) \gg (\mathcal{B}^*, \rightarrow)$ , or to “whenever  $P \mathcal{B} Q$  and  $P \rightarrow P'$ ,  $P' = Q$  or there is  $Q'$  such that  $P' \mathcal{B}^+ Q'$  and  $Q \rightarrow Q'$ ”. Indeed, if we define*

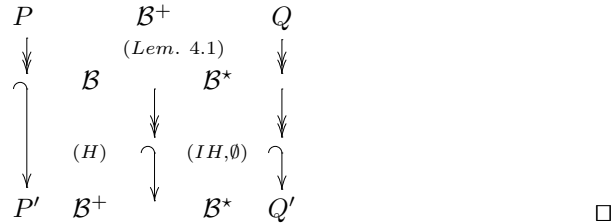
$$\begin{array}{l}
\mathcal{B} \triangleq \{ (2, 3), (3, 4), (1, 0) \} \\
\rightarrow \triangleq \{ (3, 2), (2, 1), (1, 0) \}
\end{array}
\quad
0 \xleftarrow{\mathcal{B}} 1 \leftarrow 2 \xrightarrow{\mathcal{B}} 3 \xrightarrow{\mathcal{B}} 4$$

$\mathcal{B}$  terminates and satisfies the two alternative hypotheses;  $2 \mathcal{B}^* 4$  and  $2 \rightarrow 1$ , but there is no  $i$  s.t.  $4 \rightarrow i$  and  $1 \mathcal{B}^* i$ .

A similar result: “if  $\mathcal{B}$  terminates and  $(\mathcal{B}, \rightarrow) \gg (\mathcal{B}^+, \rightarrow)$ , then  $\mathcal{B}^*$  and  $\rightarrow$  commute” is given in [9, Exercise 1.3.2]. However we are interested in showing the stronger results below, in which diagrams can be composed with other relations (this is necessary to obtain controlled simulations).

**Lemma 4.3** *Let  $\mathcal{B}, \rightarrow, \leftrightarrow$  be three relations such that  $\mathcal{B}$  terminates. If  $(\mathcal{B}, \rightarrow) \gg (\mathcal{B}^+, \rightarrow)$  and  $(\mathcal{B}, \leftrightarrow) \gg (\mathcal{B}^+, \rightarrow \leftrightarrow)$ , then  $\mathcal{B}^*$  and  $\rightarrow \leftrightarrow$  commute.*

*Proof.* As previously, we reason by well-founded induction, with the predicate  $\phi(P')$ : “For all  $P, Q$  such that  $P \rightarrow \leftrightarrow P'$  and  $P \mathcal{B}^+ Q$ , there is  $Q'$  such that  $Q \rightarrow \leftrightarrow Q'$  and  $P' \mathcal{B}^+ Q'$ ”.



**Proposition 4.4** *Let  $\mathcal{B}, \rightarrow, \leftrightarrow, \mathcal{R}, \mathcal{S}, \rightsquigarrow$  be six relations such that  $\mathcal{B}$  terminates. If  $(\mathcal{B}, \rightarrow) \gg (\mathcal{B}^+, \rightarrow)$ ,  $(\mathcal{B}, \leftrightarrow) \gg (\mathcal{B}^+, \rightarrow \leftrightarrow)$  and  $(\mathcal{R}, \rightarrow) \gg (\mathcal{B}^* \mathcal{R}, \rightarrow)$ ,  $(\mathcal{R}, \leftrightarrow) \gg (\mathcal{B}^* \mathcal{S}, \rightarrow \rightsquigarrow)$ , then  $(\mathcal{B}^* \mathcal{R}, \rightarrow \leftrightarrow) \gg (\mathcal{B}^* \mathcal{S}, \rightarrow \rightsquigarrow)$ .*

*Proof.* We use the previous lemma, within a well-founded induction. □

We can now explain the first deferred proof from the previous section: *Proof (of Theorem 3.12)*.

1. Suppose  $\mathcal{R} \rightsquigarrow \mathcal{B}^* \mathcal{R}$ , we apply Proposition 4.4, taking  $\rightsquigarrow$  for  $\rightarrow$ , and the identity relation for  $\leftrightarrow, \rightsquigarrow$ , and  $\mathcal{S}$ .

2. Suppose furthermore  $\mathcal{R} \xrightarrow{\mathcal{V}} \mathcal{S}$  and  $\mathcal{S} \xrightarrow{\mathcal{T}} \mathcal{S}$ . Lemma 4.1 ensures that  $\mathcal{B}^+$  is a silent simulation. We use Lemma 1.3(3) to close the diagram marked with a (\*) below.

$$\begin{array}{ccc}
\mathcal{B} & & \mathcal{R} \\
a \downarrow & (H) & \downarrow a \\
\mathcal{B}^+ & & \mathcal{S} \\
\tau \downarrow & (Lem. 4.1) & \downarrow \tau \\
\mathcal{B}^+ & & \mathcal{S}
\end{array}
\quad
\begin{array}{ccc}
\mathcal{R} & & \mathcal{S} \\
a \downarrow & (H) & \downarrow a \\
\mathcal{S} & & \mathcal{S} \\
\tau \downarrow & (*) & \downarrow \tau \\
\mathcal{S} & & \mathcal{S}
\end{array}$$

We then apply Proposition 4.4, using  $\mathcal{T}$  for  $\rightarrow$ ,  $\mathcal{V} \mathcal{T}$  for  $\leftrightarrow$  and  $\rightsquigarrow$ .  $\square$

## 4.2 A Generalisation of Newman's Lemma

**Lemma 4.5** *Let  $\mathcal{B}, \rightarrow, \mathcal{R}$  be three relations such that  $\mathcal{B}^+ \rightarrow^+$  terminates. If  $(\mathcal{B}, \rightarrow) \gg (\mathcal{B}^*, \twoheadrightarrow)$  and  $(\mathcal{R}, \rightarrow) \gg (\mathcal{B}^* \mathcal{R}, \twoheadrightarrow)$ , then  $\mathcal{B}^* \mathcal{R}$  and  $\twoheadrightarrow$  commute.*

*Proof.* It suffices to prove  $(\mathcal{B}^* \mathcal{R}, \twoheadrightarrow) \gg (\mathcal{B}^* \mathcal{R}, \twoheadrightarrow)$ : the commutation result then follows by a simple induction. We use an induction over the well-founded order induced by the termination of  $\mathcal{B}^+ \rightarrow^+$ , with the predicate  $\phi(P)$ : “For all  $P', Q$  such that  $P \rightarrow P'$  and  $P \mathcal{B}^* \mathcal{R} Q$ , there is  $Q'$  such that  $Q \twoheadrightarrow Q'$  and  $P' \mathcal{B}^* \mathcal{R} Q'$ ” ( $IH_1$ ). Then we do a second induction on the derivation of  $P \mathcal{B}^* \mathcal{R} Q$  ( $IH_2$ ). From the first hypothesis, we get  $P_n$  such that the leftmost diagram below holds (we show the interesting case where  $P_0 \rightarrow^+ P_n$ ). We use the internal induction to obtain  $Q_1$  in the central diagram; this is possible since any process  $P''$  such that  $P_0 \mathcal{B}^+ \rightarrow^+ P''$  satisfies  $P \mathcal{B}^+ \rightarrow^+ P''$ : the external induction hypothesis is preserved. Finally, using a third induction on the derivation  $P_1 \twoheadrightarrow P_n$ , we close the diagram by applying  $n - 1$  times the external induction hypothesis (all processes between  $P_1$  and  $P_n$  satisfy  $P \mathcal{B}^+ \rightarrow^+ P_i$ ).

$$\begin{array}{ccccccc}
P & \mathcal{B} & P_0 & \mathcal{B}^* \mathcal{R} & Q & P & \mathcal{B} & P_0 & \mathcal{B}^* \mathcal{R} & Q & P & \mathcal{B} & P_0 & \mathcal{B}^* \mathcal{R} & Q \\
\downarrow & & \downarrow & & & \downarrow & & \downarrow & (IH_2) & \downarrow & \downarrow & & \downarrow & & \downarrow \\
P' & \mathcal{B}^* & P_1 & & & P' & \mathcal{B}^* & P_1 & \mathcal{B}^* \mathcal{R} & Q_1 & P' & \mathcal{B}^* & P_1 & \mathcal{B}^* \mathcal{R} & Q_1 \\
\downarrow & & \downarrow & & & \downarrow & & \downarrow & & \downarrow & & & \downarrow & & \downarrow \\
P' & \mathcal{B}^* & P_n & & & P' & \mathcal{B}^* & P_n & & P' & \mathcal{B}^* & P_n & \mathcal{B}^* \mathcal{R} & & Q'
\end{array}$$

$\square$

By taking  $\mathcal{R} = \mathcal{I}$  in this lemma, we obtain the following corollary:

**Corollary 4.6** *Let  $\mathcal{B}, \rightarrow$  be two relations such that  $\mathcal{B}^+ \rightarrow^+$  terminates. If  $(\mathcal{B}, \rightarrow) \gg (\mathcal{B}^*, \twoheadrightarrow)$ , then  $\mathcal{B}^*$  and  $\twoheadrightarrow$  commute.*

By taking  $\mathcal{B} = \rightarrow$ , we get Newman's lemma: “Local confluence and termination entail confluence”. A different generalisation of this confluence lemma to commutation can be found in [2, Lemma 4.26]. However, the latter result is weaker than ours since it requires the termination of  $\mathcal{B} \cup \rightarrow$ , and thus the termination of both  $\mathcal{B}$  and  $\rightarrow$ .

### Remark 4.7 (up-to techniques and commuting diagrams)

The previous corollary can be proved in a direct and elegant way using the decreasing diagram techniques of Oostrom et al. [2, Theorem 4.25]:

Take  $A \triangleq \{\mathcal{B}_P/P \in \mathcal{P}\} \uplus \mathcal{P}$ , and define the following relations:

$$\begin{aligned} \succ &\triangleq \{(P, \mathcal{B}_Q) \text{ s.t. } P \mathcal{B} \rightarrow Q\} & \rightarrow_{\mathcal{B}_P} &\triangleq \{(P, Q) \text{ s.t. } P \mathcal{B} Q\} \\ &\cup \{(\mathcal{B}_P, Q) \text{ s.t. } P \rightarrow \mathcal{B}^*Q\} & \rightarrow_P &\triangleq \{(P, Q) \text{ s.t. } P \rightarrow Q\} \end{aligned}$$

However, results like Lemma 4.5 and Proposition 4.8 cannot be proved within the setting of [2], because they express properties beyond ‘pure commutation’.

Fournet [3] and others have been using results from [2] to validate up-to techniques for barbed equivalences. This is not directly comparable to the present work, since in that setting, commutation results apply directly (visible actions are not taken into account). Moreover, these works do not exploit results based on termination guarantees on the relations between processes.

**Proposition 4.8** *Let  $\mathcal{B}, \rightarrow, \mathcal{R}, \hookrightarrow, \mathcal{S}, \rightsquigarrow$  be six relations s.t.  $\mathcal{B}^+ \rightarrow^+$  terminates.*

$$\text{If } \begin{cases} (\mathcal{B}, \rightarrow) \gg (\mathcal{B}^*, \rightarrow) \\ (\mathcal{B}, \hookrightarrow) \gg (\mathcal{B}^*, \rightarrow \hookrightarrow) \end{cases} \text{ and } \begin{cases} (\mathcal{R}, \rightarrow) \gg (\mathcal{B}^*\mathcal{R}, \rightarrow) \\ (\mathcal{R}, \hookrightarrow) \gg (\mathcal{B}^*\mathcal{S}, \rightarrow \rightsquigarrow) \end{cases}$$

then  $(\mathcal{B}^*\mathcal{R}, \rightarrow \hookrightarrow) \gg (\mathcal{B}^*\mathcal{S}, \rightarrow \rightsquigarrow)$ .

*Proof.* It suffices to prove  $(\mathcal{B}^*\mathcal{R}, \hookrightarrow) \gg (\mathcal{B}^*\mathcal{S}, \rightarrow \rightsquigarrow)$ : with Lemma 4.5, this yields the expected result. Again, we use a well-founded induction over the relation  $\mathcal{B}^+ \rightarrow^+$ , with the predicate  $\phi(P)$ : “For all  $P', Q$  such that  $P \hookrightarrow P'$  and  $P \mathcal{B}^*\mathcal{R} Q$ , there is  $Q'$  such that  $Q \rightarrow \rightsquigarrow Q'$  and  $P' \mathcal{B}^*\mathcal{S} Q'$ ” ( $IH_1$ ), followed by an induction on the derivation  $P \mathcal{B}^*\mathcal{R} Q$  ( $IH_2$ ). The interesting cases are represented on the following diagrams.

$$\begin{array}{ccccc} P & \mathcal{B} & P_0 & \mathcal{B}^*\mathcal{R} & Q \\ \downarrow & & \downarrow & \downarrow & \downarrow \\ & (H) & \downarrow & \mathcal{B}^*\mathcal{R} & \downarrow \\ & & & & (IH_1) \\ & & \downarrow & & \downarrow \\ P' & \mathcal{B}^* & P'_0 & \mathcal{B}^*\mathcal{S} & Q' \end{array} \quad \begin{array}{ccccc} P & \mathcal{B} & P_0 & \mathcal{B}^*\mathcal{R} & Q \\ \downarrow & & \downarrow & \downarrow & \downarrow \\ & (H) & \downarrow & \mathcal{B}^*\mathcal{S} & \downarrow \\ P' & \mathcal{B}^* & P'_0 & \mathcal{B}^*\mathcal{S} & Q' \end{array} \quad \square$$

Like in the proof of Theorem 3.12, we use Proposition 4.8 and Corollary 4.6 to establish Theorem 3.13.

**A theorem prover formalisation of our results.** The proofs about diagrams sometimes require non trivial inductive reasoning, and it is easy to make mistakes when nesting several inductions. These results, as well as all proofs in the paper – except for Subsection 2.3 – have been formally checked in the Coq proof assistant [6], and the descriptions of the proofs we give actually closely follow the proof scripts. These developments are available from [5].

## 5 Applications

### 5.1 Correctness Proof for an Optimised Abstract Machine

We present an example from [4], where we study an abstract machine for the distributed execution of Safe Ambients. We improve a previously existing machine by introducing garbage collection mechanisms for *forwarders*: agents that are created to retransmit messages along the net upon code migration.

The optimised abstract machine is proved correct by establishing a bisimilarity result with the original machine. Due to the failure of existing up-to techniques, correctness is established by a direct bisimulation argument in [4]. We explain why up to expansion cannot be used in [4] to obtain a better proof, and we show how this can be improved using our techniques.

We have presented in a setting where the processes being compared belong to a single set, and where the transition relation is the same on both sides. Our developments can be straightforwardly generalised to the case where processes and transitions are taken from two distinct LTS sharing the same set of actions.

We define two LTS  $(\mathcal{P}_1, \mathcal{L}, \rightarrow_1)$  and  $(\mathcal{P}_2, \mathcal{L}, \rightarrow_2)$ , corresponding respectively to the optimised and original versions of the system. We denote by  $\xrightarrow{\alpha_1}$  and  $\xrightarrow{\alpha_2}$  the respective labelled transitions; we write  $\approx_i, \succsim_i, \approx\approx_i$  for the corresponding relations defined within  $(\mathcal{P}_i, \mathcal{L}, \rightarrow_i)$ ;  $\approx_1 \approx_2$  stands for the bisimilarity relation between the two systems. Given a set of *names*  $h, k$ , and a set of *messages*  $M, N, \dots, S$ , we define states, also called *nets*  $(U, V)$ , and *actions*  $(\alpha \in \mathcal{L})$ , as follows:

$$\begin{array}{llll}
U, V := & U \mid V & (\text{parallel composition}) & \alpha := \bar{h}\{M\} & (\text{emission}) \\
& | & h \square k & | & h\{M_k\} & (\text{reception}) \\
& | & h \triangleright k & | & h \triangleright k & (\text{forwarder}) \\
& | & h \triangleleft k & | & \tau & (\text{silent action}) \\
& | & h\{M_k, l\} & | & & (\text{message}) \\
& | & h\{\rightsquigarrow k\} & | & l := \square \mid h::l & (\text{relocation message})
\end{array}$$

A net consists in a set of *agents*: sites or forwarders, that communicate using messages. We say that the *name* (resp. *parent*) of an agent  $h \square k, h \triangleright k$  or  $h \triangleleft k$  is  $h$  (resp.  $k$ ). The *parent relation* of a net is the binary relation over names such that  $h$  is related to  $k$  iff there is an agent named  $h$  having  $k$  as parent. Finally, we define *processes* as nets for which the induced parent relation forms a tree, and in which messages appear in a coherent way w.r.t. the transition rules defined in the sequel (we refer to [4] for a more formal account of this notion of well-formedness).  $\mathcal{P}_1$ , the set of *optimised processes*, is the set of such well-formed nets;  $\mathcal{P}_2$ , the set of *original processes*, is the subset of  $\mathcal{P}_1$  such that each process contains no blocked forwarder and no relocation message.

The two labelled relations, over  $\mathcal{P}_1$  and  $\mathcal{P}_2$ , are defined modulo associativity and commutativity of the parallel composition operator by the following rules:

$$\begin{array}{llll}
U \mid h \square k & \xrightarrow{h \triangleright k}_{1,2} & U \mid h \triangleright k & \\
U \mid h \square k & \xrightarrow{\bar{h}\{M\}}_{1,2} & U \mid h \square k \mid k\{M_h, \square\} & \\
U \mid h \square k \mid h\{M_{h'}, \square\} & \xrightarrow{h\{M_{h'}\}}_{1,2} & U \mid h \square k & \\
U \mid h \triangleright k \mid h\{M_{h'}, \square\} & \xrightarrow{\tau_1} & U \mid h \triangleright k \mid k\{M_{h'}, \square\} & \\
U \mid h \triangleright k \mid h\{M_{h'}, l\} & \xrightarrow{\tau_2} & U \mid h \triangleleft k \mid k\{M_{h'}, h::l\} & \\
U \mid h \square k \mid h\{M_{h'}, k'::l\} & \xrightarrow{\tau_2} & U \mid h \square k \mid h\{M_{h'}, l\} \mid k'\{\rightsquigarrow h\} & \\
U \mid h \triangleleft k \mid h\{\rightsquigarrow k'\} & \xrightarrow{\tau_2} & U \mid h \triangleright k' &
\end{array}$$

We briefly comment on the rules. Both systems share the same visible transitions: the first rule lets a site evolve into a forwarder; the second rule describes the emission of a message  $M$  from site  $h$ ; the last rule corresponds to the reception of message  $M$ , sent by site  $h'$  to site  $h$ .

The silent part for the original system ( $\xrightarrow{\tau_1}$ ) defines the simple transmission of messages by forwarders. Message transmission is optimised in the second LTS by contracting forwarder chains following the classical approach of Tarjan sets. In the first  $\xrightarrow{\tau_2}$  transition, a message is forwarded, the forwarder enters blocked state, and its name is stored into the list carried by the message. In the second rule, a site sends a relocation message to each forwarder registered in the (nonempty) transmitted list — when the list is empty, a visible transition is triggered. In the last  $\xrightarrow{\tau_2}$  rule, a blocked forwarder is relocated and resumes computation.

We define the following relations;  $\mathcal{E}_1$  and  $\mathcal{E}_2$  correspond to the erasure of a forwarder in a process:

$$\begin{aligned} \mathcal{E}_1 &\triangleq \xrightarrow{\tau_1} \mathcal{E}'_1 \xrightarrow{\tau_1^{-1}} \\ \mathcal{E}'_1 &\triangleq \{ (h \triangleright k \mid U, U\{h \setminus k\}) \} \cap \mathcal{P}_{1\downarrow} \times \mathcal{P}_{1\downarrow} & T_1 &\triangleq \xrightarrow{\tau_1} \cup \mathcal{E}_1 \\ \mathcal{E}_2 &\triangleq \{ (h \triangleright k \mid U, U\{h \setminus k\}) \} \cap \mathcal{P}_2 \times \mathcal{P}_2 & T_2 &\triangleq \xrightarrow{\tau_2} \cup \mathcal{E}_2 \end{aligned}$$

where  $\{h \setminus k\}$  denotes the textual replacement of  $h$  by  $k$ , and  $\mathcal{P}_{1\downarrow}$  is the set of processes in normal form w.r.t.  $\xrightarrow{\tau_1}$ .  $\mathcal{E}_1$  is defined *modulo silent transitions* in order to handle easily the erasure of a blocked forwarder.

**Lemma 5.1**  *$T_1$  and  $T_2$  are terminating and locally confluent relations.*

We write  $P_T$  for the normal form of  $P$  w.r.t.  $T_1$  or  $T_2$ , depending on the context. Notice that the normal form of an optimised process is an original process without forwarders.

**Lemma 5.2**  *$T_2$  is an expansion;  $T_1$  is a controlled bisimulation.*

*Proof.* We show easily that both  $\xrightarrow{\tau_2}$  and  $\mathcal{E}_2$  are expansions. We prove that  $T_1$  is a controlled bisimulation:

- First we show that  $\xrightarrow{\tau_1}$  is a bisimulation:  $\xrightarrow{\tau_1}$  evolves to  $\xrightarrow{\tau_1}$ ,  $\xrightarrow{\tau_1^+ \tau_1^+}$  terminates. We can thus apply Theorem 3.13 to  $\xrightarrow{\tau_1}$ : it is a controlled relation, and  $\xrightarrow{\tau_1}$  is a simulation. Since furthermore  $\xrightarrow{\tau_1^{-1}}$  evolves to the identity relation,  $\xrightarrow{\tau_1}$  is a bisimulation.
- Then we show that  $\mathcal{E}_1$  is a bisimulation: define  $\mathcal{F}(\mathcal{R}) \triangleq \xrightarrow{\tau_1} \mathcal{R} \xrightarrow{\tau_1^{-1}}$ .  $\mathcal{F}$  is a weakly monotonic function since  $\xrightarrow{\tau_1}$  is a bisimulation. It is easy to see that  $\mathcal{E}'_1$  evolves silently to itself, since its elements are in normal form w.r.t.  $\xrightarrow{\tau_1}$ . We check that  $\mathcal{E}'_1$  evolves visibly to  $\mathcal{F}(\mathcal{E}'_1)$ , so that using Proposition 2.2,  $\mathcal{F}^*(\mathcal{E}'_1) = \mathcal{E}_1$  is a simulation. Similarly, we obtain that  $\mathcal{E}_1^{-1} = \mathcal{F}^*(\mathcal{E}'_1^{-1})$  is a simulation, so that  $\mathcal{E}_1$  is a bisimulation.
- These results ensure that  $T_1$  evolves to  $T_1^*$ ; since  $T_1^+ \xrightarrow{\tau_1^+} \subseteq T_1^+$  terminates, we can apply Theorem 3.13:  $T_1$  is a controlled relation and finally it is a controlled bisimulation.  $\square$

**Remark 5.3 (when expansion cannot be used)**

$T_1$ , like  $\xrightarrow{\tau_1}$  and  $\mathcal{E}_1$ , is not contained in  $\approx$  (and hence neither in  $\approx$ ): define

$$\begin{aligned} P &= h\{M, []\} \mid h\{N, []\} \mid h \triangleright k \mid k \square k' \\ Q &= h\{N, []\} \mid h \triangleleft k \mid k\{M, h\} \mid k \square k' \end{aligned}$$

( $P$  moves to  $Q$  by transmitting  $M$  along the forwarder). After two silent steps,  $P$  may do the visible action  $k\{N\}$ , however  $Q$  has to do at least four silent actions before being able to answer with the same visible action: it has to unblock its forwarder first. Therefore  $P \not\lesssim Q$ . A formal proof corresponding to  $\mathcal{E}_1 \not\lesssim \mathcal{E}_2$  can be found in [4].

Lemma 5.2 provides powerful up-to techniques: in both systems we can use normalised candidate relations: whenever a forwarder is created, it can be erased using  $\mathcal{E}$ . Whenever a transition leads to a possible sequence of silent actions, these can be executed immediately, in order to get back to a normalised process.

As shown by the following theorem, this is actually useful in our setting, since normalised processes of both systems can be compared syntactically (notice that since the labelled transitions are distinct,  $P_T \approx_2 Q_T$  is not a trivial consequence of the equality  $P_T = Q_T$ ):

**Theorem 5.4** *Let  $P \in \mathcal{P}_1$  and  $Q \in \mathcal{P}_2$ .*

*If  $P_T = Q_T$  (syntactically), then  $P \approx_2 Q$ .*

*Proof.* Define  $\mathcal{R} \triangleq \{ (U, U) \mid U \text{ is normalised} \} \subseteq \mathcal{P}_1 \times \mathcal{P}_2$ . We check that  $\mathcal{R}$  evolves silently to  $\mathcal{R}$  (no silent evolution), and visibly to  $T_1^* \mathcal{R} T_2^{-1*}$ , and thus, using Lemma 5.2, to  $\approx_1 \mathcal{R} \approx_2$ . Similarly, the same properties hold for  $\mathcal{R}^{-1}$ , so that by Theorem 2.5,  $\mathcal{R} \subseteq \approx_2$ . Finally, we have  $P \approx_1 P_T \approx_2 Q_T \approx_2 Q$ .  $\square$

## 5.2 Validating a Caching Technique

We now sketch the analysis of a caching technique. We study a system whose purpose is to serve *requests* for information by giving appropriate *answers*. In the simple version, the system accepts a request, computes the corresponding answer (for example, by searching for it in a database), and returns it. The optimised version of the system maintains a *cache*, in which previously computed answers can be stored, as well as some answers that the system might want to compute in advance (e.g., one could think of predicting requests that are deemed to be liable, in view of previous sessions). Within a large proof of some property of the system, we would like to be able to reason *up to the cache*, that is, consider a candidate relation where all processes have an empty cache.

Processes are pairs  $\langle R \parallel C \rangle$ , where  $R$  is a set of *requests*, and  $C$  is a set of *cached values*. In both cases, we denote by  $x::S$  the addition of an element  $x$  to a set  $S$ , and by  $\#S$  the size of the set  $S$ . The rules are given below:

$$\begin{array}{ccc} \langle R \parallel C \rangle & \xrightarrow{a_r} & \langle r::R \parallel C \rangle & \langle r::R \parallel r::C \rangle & \xrightarrow{b_r} & \langle R \parallel r::C \rangle \\ \langle R \parallel C \rangle & \xrightarrow{\tau} & \langle R \parallel r::C \rangle & \langle R \parallel s::C \rangle & \xrightarrow{\tau} & \langle R \parallel r::C \rangle \end{array}$$

The first visible action,  $a_r$ , is the reception of a request  $r$ . The two transitions at the bottom are silent: they show how a value can be added or replaced in the cache. Once the answer to this request is available, it can be sent using the second visible action,  $b_r$ .

It can be proved that for any set of requests  $R$ ,  $\langle R \parallel C \rangle \approx \langle R \parallel \emptyset \rangle$ . However, this is not sufficient to obtain an up-to technique: the cache is filled using silent actions, for which reasoning up to bisimilarity is not allowed. It does neither hold that  $\langle R \parallel C \rangle \lesssim \langle R \parallel \emptyset \rangle$  for any  $R$ : if  $R$  and  $C$  both contain a value  $r$ ,

$\langle R \parallel C \rangle$  can do a visible action by completing the request  $r$ , while  $\langle R \parallel \emptyset \rangle$  has to compute  $r$  before being able to do the corresponding visible action.

With Theorem 3.12, we prove that  $\mathcal{B} \triangleq \{ (\langle R \parallel C \rangle, \langle R \parallel C' \rangle) \text{ s.t. } \#C' < \#C \}$  is a controlled bisimulation, which gives a way to reason ‘up to the cache’.

## 6 Concluding remarks

The framework we have introduced is based on a separation between visible and silent transitions, and provides flexible and powerful proof techniques, that depend on the kind of transition they are applied to. For instance, reasoning up to weak bisimilarity is possible as long as it is restricted to visible actions. Experience on case studies (such as the one in [4]) has to be developed in order to have a better understanding of how our techniques can be best combined, and how to tune the distinction between visible and internal computation steps.

Due to the presence of labelled transitions, results about decreasing diagrams from [2] are not applicable directly in our setting. We plan to study how the theory of decreasing diagrams can be adapted to keep track of visible actions. This could be a way to provide an abstract approach for the definition of ‘up to transitivity’ techniques based on termination guarantees.

**Acknowledgements.** We would like to thank Davide Sangiorgi for his comments and suggestions, and Daniel Hirschhoff for helpful discussions and a great help during the redaction process.

## References

- [1] S. Arun-Kumar and M. Hennessy. An efficiency preorder for processes. *Acta Informatica*, 29(9):737–760, 1992.
- [2] M. Bezem, J. W. Klop, and V. van Oostrom. Diagram techniques for confluence. *Information and Computation*, 141(2):172–204, 1998.
- [3] C. Fournet. *The Join-Calculus: a Calculus for Distributed Mobile Programming*. PhD thesis, Ecole Polytechnique, 1998.
- [4] D. Hirschhoff, D. Pous, and D. Sangiorgi. An Efficient Abstract Machine for Safe Ambients. Technical Report 2004–63, LIP – ENS Lyon, 2004. An extended abstract appeared in the proceedings of COORDINATION’05.
- [5] D. Pous. Web appendix of this paper, 2005. Available at <http://perso.ens-lyon.fr/damien.pous/upto>.
- [6] INRIA projet Logical. The Coq proof assistant. <http://coq.inria.fr/>.
- [7] D. Sangiorgi. On the Bisimulation Proof Method. *Mathematical Structures in Computer Science*, 8:447–479, 1998.
- [8] D. Sangiorgi and R. Milner. The problem of “Weak Bisimulation up to”. In *Proc. CONCUR ’92*, volume 630 of *Lecture Notes in Computer Science*, pages 32–46. Springer Verlag, 1992.
- [9] TeReSe. *Term Rewriting Systems*. Cambridge University Press, 2003.